

Chapter: 3.0 Initiation and Planning Stage

Description: This is the first stage in the lifecycle of an information systems engineering project. Project Management Methodology (PMM) and System Engineering Methodology are tightly integrated. It is imperative that the project manager ensures participation of the business client in the creation of the PMM documents.

The majority of the outputs produced during this stage are Project Management Methodology (PMM) documents, such as the Business Case, the Project Charter, the Project Plan and the Quality Management Plan. The SEM Software Configuration Management Plan and the Maintenance Plan are initiated during this stage.

Although the following italicized paragraphs focus on PMM processes, a solid project management foundation is necessary for SEM deliverables and subsequent stages.

Project planning applies to all projects regardless of their size. Planning involves selecting the strategies, policies, programs, and procedures for achieving the objectives and goals of the project. Planning is deciding, in advance, what to do, how to do it, when to do it, where to do it, and who is going to do it.

The requirements identified in project related materials, (e.g., a business case document, are the primary input to the Project Plan.) The level of detail will vary depending on project size. The preparation of the Project Plan and related materials involves several critical planning issues such as the identification of preliminary requirements; staff, schedule, and cost estimates; the technical and managerial approaches that will be used; and the assessment of potential risks associated with the project. This information forms the foundation for all subsequent planning activities.

During this stage, the system owner and users are interviewed to: identify their business needs and expectations for the product; gain a common understanding of the task assignment; and determine how the project supports the State of Michigan's long-range information resource management plans. The system owner is the enterprise unit that is funding the project, and users are the State of Michigan employees and contractors who will use the product.

In this stage, the project team should be focused on identifying what the project will automate, and whether developing an IT solution makes sense from business, cost, and technical perspectives. If the project is feasible, then time, cost, and resource estimates must be formulated for the project, and risk factors must be assessed. It is important for the project team to work closely with representatives from all functional areas that will be involved in providing resources, information, or support services for the project (see Touch Points). The information gathered in this stage is used to plan and manage the project throughout its lifecycle.

This stage involves development of a Software Configuration Management Plan (SEM) to track and control work products and a Quality Management Plan (PMM) to assure the production and operation of high quality products on schedule, within budget, and within the constraints specified by the system owner and user.

Input:

The following items provide input to this stage:

- Requirements identified in project related materials, (e.g., a business case)
- Related project initiation materials

High-Level Activities:

The remainder of this chapter is divided into sections that describe specific high-level activities performed during this stage. These activities represent the minimum requirements for a large information systems engineering effort. Notes are provided, as applicable, to assist in customizing these lifecycle stage requirements to accommodate different size efforts. The high-level activities are presented in the sections listed below.

- 3.1 Develop Software Configuration Management Plan
- 3.2 Develop Maintenance Plan

Touch Points:

The following touch points are involved in the Initiation and Planning Stage:

- Contracts and Procurement
 - Assignment of a Contract Liaison if procuring goods or services
 - Completion on DIT-0153 Bid Information Sheet if procuring goods or services
- Enterprise Architecture (EA)
 - Review relevant EA materials (e.g., roadmaps, solution patterns)
 - Develop EA Solution Assessment for each alternative (refer to Appendix C for assistance in developing the EA Solution Assessment.
- Security
 - Notify your Security Liaison of project initiation
 - Review MDIT and Agency Security Policies
 - Initiate Security Plan, including Data Classification and System Criticality sections
- Other
 - Initiate Business Continuity Planning process (DMB has a website for this purpose.)

Output:

Several work products are developed during this stage. The work products listed below are the minimum requirements for a large project. Deviations in the content and delivery of these work products are determined by the size and complexity of a project. Explanations of the work products are provided under the applicable activities described either in the remainder of this chapter or in the PMM.

SEM Templates:

- Software Configuration Management Plan (*initial*)
- Maintenance Plan (*initial*)

PMM Templates:

- Business Case
- Concept Document (i.e., Feasibility Study)
- Project Charter
- Project Plan (includes Quality Management Plan)
- Security Plan (*initial*)

Other Outputs:

- Enterprise Architecture (EA) Solution Assessment for each potential solution option (*initial*)
- Business Continuity Plan (*initial*)

A diagram showing the work products associated with each SEM stage is provided in *Exhibit 3.0-1, SEM Overview – Initiation and Planning Stage Highlighted*. The activities for this stage are emphasized in bold.

Review Process:

Quality reviews are necessary during this stage to validate the project and associated work products. The activities that are appropriate for quality reviews are identified in this chapter and chapter 2.0, Lifecycle Model. The time and resources needed to conduct the quality reviews should be reflected in the project resources, schedule, and work breakdown structure.

Structured Walkthrough (SWT)

Requirements for a peer review or a more formal structured walkthrough are documented under *Review Process* at the end of each Task, Subtask, or Activity section in this stage. The State of Michigan guide titled *Structured Walkthrough Process Guide* provides a procedure and sample forms that can be used for SWTs. This document is available on the MDIT SUITE website.

Stage Exit

Schedule a Stage Exit as the last activity of the Initiation and Planning Stage to enable the project approvers to review project deliverables and provide a concur/non-concur position to the project manager. The State of Michigan guide titled *Stage Exit Process Guide* provides a procedure and sample report form that can be used for stage exits. This document is available on the MDIT SUITE website.

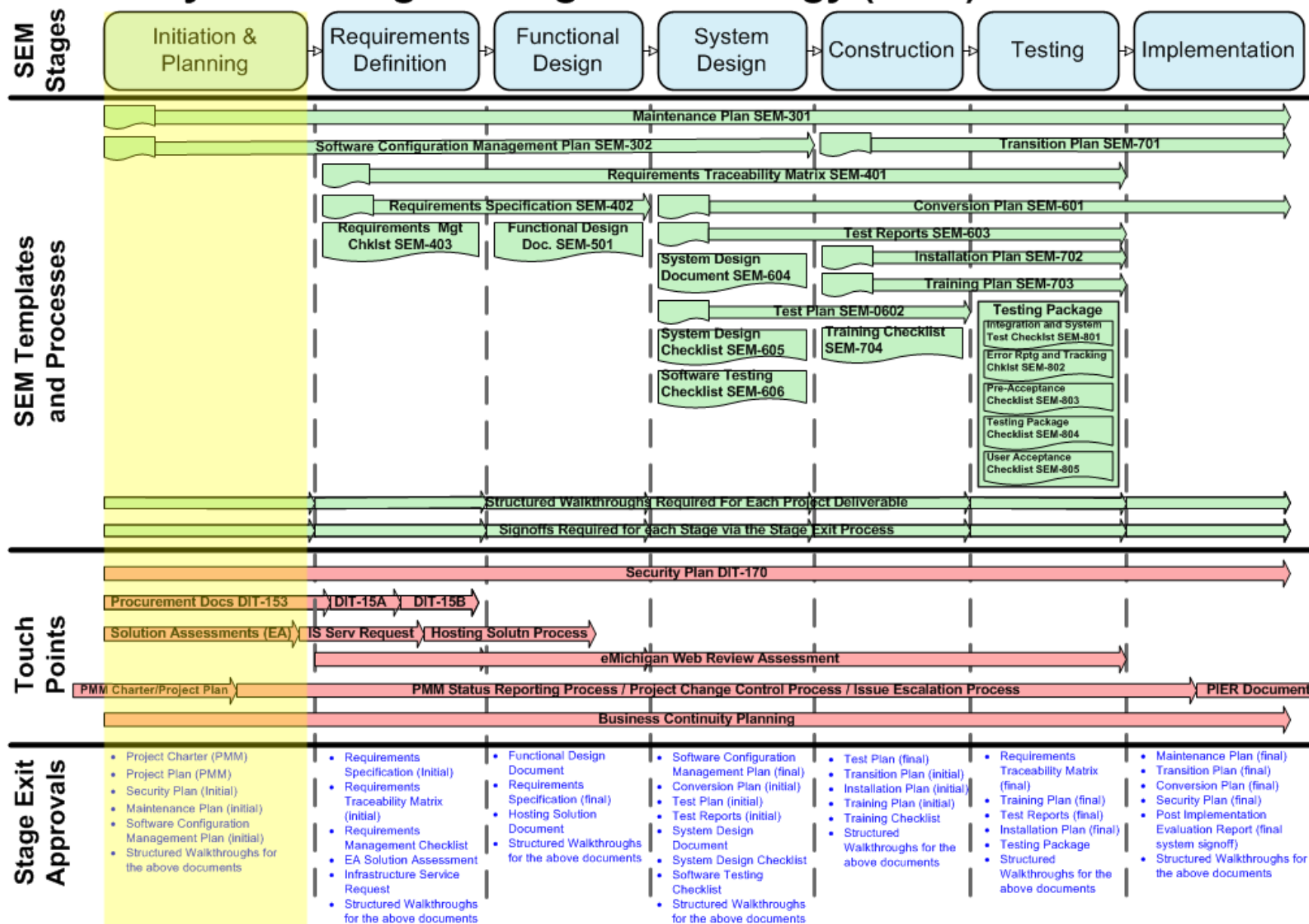
References: Chapter 2.0, Lifecycle Model, *Quality Reviews*, provides an overview of the Quality Reviews to be conducted on a project.

Bibliography: The following materials were referenced in the preparation of this chapter.

1. Carnegie Mellon University, Software Engineering Institute, *Capability Maturity Model: Guidelines for Improving the Software Process*, Addison Wesley Longman, Inc., 1994.
2. Project Management Institute, *A Guide to the Project Management Body of Knowledge*, Pennsylvania, 1996.
3. The Institute of Electrical and Electronics Engineers, Inc., *IEEE Standard for Developing Software Lifecycle Processes*, IEEE Std 1074-1991, New York, 1992.
4. U.S. Department of Energy, *Departmental Information Systems Engineering: Volume 1, Information Systems Engineering Lifecycle*, September 2000.
5. U.S. Department of Energy, *DOE/NV Software Management Plan*, Nevada Operations Office, May 1991.
6. U.S. Department of Energy, *Software Management Guide*, DOE/AD-0028, 1992.

Exhibit 3.0-1 SEM Overview Diagram – Initiation and Planning Stage Highlighted

Systems Engineering Methodology (SEM) Overview



Activity: 3.1 Develop Software Configuration Management Plan

Responsibility: Project Manager or Software Configuration Manager

Description: Software configuration management (SCM) is a standard and consistent set of processes used to control version changes to work products and other artifacts during all stages of the project lifecycle. The goals of configuration management are to identify the configuration of the product (i.e., work products and their descriptions) at given points in time, to systematically control changes to the configuration, and to maintain the integrity and traceability of the configuration throughout the lifecycle.

Software configuration management includes the following activities:

- An SCM plan is prepared for each project according to a documented process and is coordinated with affected groups and individuals.
- A documented and approved software configuration management plan is used as the basis for performing the software configuration management activities.
- Authority for managing the project's baselines is established (e.g., a Software Configuration Control Board - SCCB).
- An SCM library system is established as a repository for the baselines.
- Work products are identified and placed under configuration control.
- Change requests and problem reports for all items/units are initiated, recorded, reviewed, approved, and tracked according to a documented process.
- Changes to baselines are controlled according to a documented process.
- Products from the baseline library are created and their release is controlled according to a documented procedure.
- Status of items/units is recorded according to a documented process.
- Standard reports documenting the SCM activities and the contents of the baseline are developed and made available to affected groups and individuals.
- Baseline audits are conducted according to a documented process.

The work products placed under SCM include the products that are delivered to the customer (e.g., the requirements documents and the source code) and the items that are identified with or required to create these products (e.g., the compiler). A baseline library is established containing the baselines of the configuration items as they are developed. Changes to baselines and the release of products built from the baseline library are systematically controlled via the change control and configuration auditing functions of configuration management.

The Software Configuration Manager (or the individual assigned configuration responsibilities) is responsible for routine evaluation of the product. The Software Configuration Manager controls changes that are introduced into the systems product environment. The Software Configuration Manager is responsible for the processes necessary to correct faults in the environment and product. The Software Configuration Manager is not responsible for any overall project deadlines or management issues.

Work Product:

An SCM plan that defines the configuration management policies and procedures is required for each project. The plan is developed early in the lifecycle to ensure the control of changes as soon as the project requirements are approved and baselined. In this stage, the plan addresses activities that are platform independent, such as identifying the items that will be placed under configuration management. As the project progresses through the lifecycle stages, the plan is expanded to reflect platform specific activities.

The SCM plan addresses the following types of responsibilities and activities:

- Defining the required configuration management policy and procedures.
- Maintaining all documents in an easily accessible central library.
- Receiving unit test and integrated builds and related documentation from the developer.
- Performing version control procedures on unit and integrated builds received from the developer.
- Packaging a tested unit or integrated build for return to developer or production as required.
- Shipping an approved unit or integrated build to production as required.
- Maintaining an archive of project-related correspondence between members of the project team.
- Overseeing the release and subsequent distribution of configuration items.

Provide enough information in the plan so that compliance can be monitored by means of project records. Whenever feasible, acquire automated SCM tools to check compliance with project standards, the validity and consistency of product design, requirements, and system performance.

Based on the complexity of the project and the anticipated volume of changes, a Software Configuration Management Plan can be developed for a specific project, an existing plan can be modified to suit the requirements of a project, or a plan can be developed to manage all of the projects supporting a particular system owner's organization. Place a copy of the SCM Plan in the Project File.

Review Process: Conduct structured walkthroughs to validate that the configuration management approach, the configuration identification, change control, status accounting, and auditing procedures are appropriate for the project.

Resources: The MDIT SUITE website contains a sample SCM plan and template.

The Institute of Electrical and Electronics Engineers (IEEE) Standard for Software Configuration Management Plans (Std 828-1990) provided guidance on developing these plans.

Activity: 3.2 Develop Maintenance Plan

Responsibility: Project Manager

Description: The purpose of the Maintenance Plan is to determine the scope of the maintenance effort, identify the process and tools, quantify the maintenance effort (personnel and resources), and identify anticipated maintenance requirements. The Maintenance Plan needs to define the maintenance process and its boundaries or scope. The maintenance process beginning point should be defined (e.g., receipt of a change request or planned COTS version upgrade) and the ending action should be defined (e.g., delivery and sign-off of a product). The process is a natural outgrowth of many of the software configuration management procedures. A description of the overall flow of work within the maintenance process should be included. The maintenance process can be tailored to the type of maintenance being performed and can be divided in several different ways. This can include different processes for corrections or enhancements or small or large changes.

The maintenance requirements need to be identified and quantified (sized) in the Maintenance Plan to determine the future maintenance load for the organization. The following issues should be considered when defining the requirements.

- Expected external or regulatory changes to the product
- Expected internal changes to support new requirements
- Requirements deferred from current project to later release
- Wish-list of new functions and features
- Expected upgrades for performance, adaptability, or connectivity
- New lines of business that need to be supported
- New technologies that need to be incorporated

The requirements for the maintenance staff also need to be established. At this stage, the maintenance plan should begin to address the following:

- Number of maintainers, their job descriptions, and required skill levels
- Experience level of the maintenance staff
- Documented maintenance processes at the systems and program levels
- Actual methods used by development staff
- Tools used to support the maintenance process
- Current work load and estimates of future needs

**Description,
continued:**

An important part of the maintenance plan is an analysis of the hardware and software most appropriate to support the maintenance organization's needs. The development, maintenance, and test platforms should be defined and differences between the environments described. Tools sets that enhance productivity should be identified and provided. Tools should be accessible to all who need them, and sufficient training provided so that their use will be well understood.

Although all systems need maintenance, there comes a time when maintenance is no longer technically or fiscally viable. Issues such as resources, funds, and priorities may dictate that a system should be replaced rather than changed. The maintenance plan should identify the criteria that indicate the product is ready for retirement or replacement, such as the failure rate, age of code, and incompatibility with current technology.

Work Product:

Initiate development of the Maintenance Plan. Place a copy of the draft Maintenance Plan in the Project File. As part of the Implementation Stage, once system installation and acceptance are complete, determine if the Maintenance Plan needs to be revised, if so, update and finalize the Maintenance Plan. Place a copy of the initial Maintenance Plan in the Project File.

Review Process:

Conduct a structured walkthrough to ensure that the Maintenance Plan accurately reflects the necessary information.

The Maintenance Plan is formally reviewed during the Stage Exit process.

Reference:

Refer to the forthcoming *MDIT Systems Maintenance Guide*, for more information on maintaining the product.

Resource:

A template for the Maintenance Plan is available on the MDIT SUITE website.